

**Review Article**

# Evaluating Software System Reliability Using Architecture Based Approach

**Sabbineni Srinivas Rao, Inuganti Nava Sahitha, Godithi Sireesha, Palem Manoj**

Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India

**Email address:**

srinu1479cse@kluniversity (S. S. Rao), sahitha.inuganti11@gmail.com (I. N. Sahitha),

sireeshagodithi7@gmail.com (G. Sireesha), manoj33p@gmail.com (P. Manoj)

**To cite this article:**Sabbineni Srinivas Rao, Inuganti Nava Sahitha, Godithi Sireesha, Palem Manoj. Evaluating Software System Reliability Using Architecture Based Approach. *International Journal of Intelligent Information Systems*. Vol. 7, No. 1, 2018, pp. 1-4. doi: 10.11648/j.ijis.20180701.11**Received:** January 20, 2018; **Accepted:** February 5, 2018; **Published:** February 23, 2018

---

**Abstract:** Programming dependability is those failure-free programming operations for a specified time clinched alongside a specified earth. On acquire secondary unwavering quality to expansive what's more intricate framework, utilize architecture-based approach. Software reliability is one of the major attributes of the software quality attributes that are availability, interoperability, maintainability, manageability, performance, reliability, reusability. To obtain reliability, used mainly fault tolerance mechanisms in the design process. In this paper there is a comparison between error recovery along with fault tolerance mechanisms versus error propagation in evaluating software system reliability. Here compared two case studies which produce the software reliability.

**Keywords:** Software Reliability, Software Architecture, Fault Tolerance, Error Propagation

---

## 1. Introduction

These days mossy cup oak of the programming frameworks would utilized within incredulous situations which need aid intricate should tackle the genuine cases which prompt disasters going starting with budgetary misfortune to debilitate mankind's exists. These disasters are due to the unreliability of the software system. To obtain the reliability of the software system customary methodologies like black box trying will programming unwavering quality demonstrating will be connected in the internal structure, be that as with the advancement of the component-based paradigm, these are not suitable.

A standout amongst the principle objectives about architecture-based dependability deliberations is estimating those frameworks unwavering quality toward leveraging disappointment determining systems. The existing construction modeling built models need aid primarily arrangement. Clinched alongside general, those state-based methodologies model a programming framework. Eventually Tom's perusing mapping the probabilistic control stream chart should a state space model. These models portraying the framework at those

constructions modeling level incorporates Discrete-Time Markov chain, a nonstop occasion when Markov chain alternately a Semi-Markov transforms [1]. The path-based models take after the time permits execution ways of the architecture; in any case they are not suitability for cyclic structural engineering because of limitless way [1]. Propagation-based models concentrate on slip proliferation around segments which representable the likelihood about dependencies from claiming part disappointments. [2].

Those principle varieties about the work, that recognizes it starting with a large portion of the existing systematic approaches, comprises done acknowledging a paramount structural aspect, those slip proliferation from part should part. Neglecting this angle might lead, at those best, on excessively negative predictions of the framework reliability, that might foundation unnecessary plan and execution endeavors should enhance it. On unwavering quality dissection may be used to drive those determination of components, it might prompt bad estimates of the unwavering quality of different part assemblies, hence bringing on those determination for a gathering which may be lesquerella dependable over others.

Key contributions of this paper are firstly estimated the probabilities of the error recovery system using the fault

tolerance mechanisms and secondly probabilities of the system with error propagation.

## 2. Related Work

In this survey, there is a detailed explanation on software architecture and fault tolerance mechanisms. For many years studied the architecture based approach and during the Gokhale [2] defines the work into five categories. They are modeling, analysis, parameter estimation, validation and optimization. But the architecture based approach is purely based on the modeling that is at the design phase. Here mostly removed the errors or faults at the time of implementation or testing which leads to the scope loss. When the errors are found at the time of the design then the reliable software is obtained and can reduce the time loss.

In architecture-based approach, there is reliability analysis which can be calculated through different techniques. They are state based [3][4], path-based [6], propagation based[7], and supplementary techniques.

For programming unwavering quality engineering, there are four primary methodologies should expand framework reliability, which will be shortcoming prevention, flaw line removal, issue tolerance, and issue determining. Since source-codes and internal plans about web benefits would inaccessibility will administration users, it is being troublesome to utilize issue aversion and deficiency evacuation systems should raise fault-free service-oriented frameworks. An alternate methodology for building dependable systems, product flaw line tolerance, makes the framework stronger. Eventually Tom's perusing masker faults as opposed to uprooting faults.

Person approach on programming issue tolerance, otherwise called design diversity, will be should utilize functionally equal yet freely intended parts on endure faults.

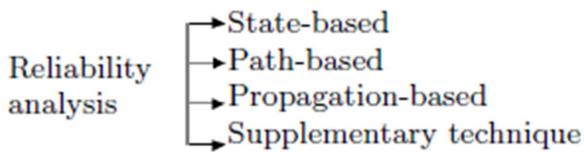


Figure 1. Techniques of reliability analysis.

In this paper, the usage of state-based approach for architecture based analysis gives rise to architectural model and component failure model [5]. A structural model may be an outline made utilizing accessible standards, previously, which the elementary concern is with show a situated from claiming tradeoffs intrinsic in the structure what's more configuration of a system, it is used to take sentiment starting with the look client by the programming modeler.

Component failure model is a mechanism where in the software system when one component is failed then also the system does not fail and continue until reaching the output. These two models are combined and gives rise to two methods called hierarchical method and composite method to analyze the reliability.

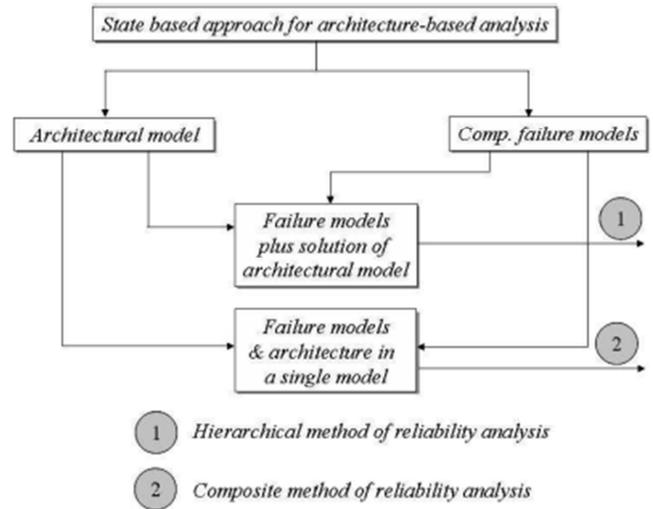


Figure 2. Analysis methods in state based approach [1].

## 3. Problem Description

Threads to reliability can occur from failure error and fault through which fault tolerance is associated. Despite all efforts, there are many faults during testing. These faults produce errors when they are activated. There are two types of faults, active and inactive faults. A dynamic flaw line may be whichever inner issue alternately outside fault, yet not at faults prompt errors. Deficiency tolerance components need aid of two types, slip handling, also deficiency taking care of. That fundamental system about lapse taking care of may be lapse disposal. Those fundamental components from claiming issue taking care of maybe it worries with following the establishes of faults. Deficiency tolerance may be a survival quality which empowers an arrangement to experience disappointments also conveys constant administration should clients. For deficiency tolerance, a large portion utilized usage strategy will be outlined differences.

Slip proliferation may be the transform from claiming deciding those questionable matters for a response got from a figuring. Each duration of the time information will be measured, there is a vulnerability connected with that estimation. There are three components which are error-free, faulty and the failed states in the reliability analysis considering error propagation. Suppose there is a fault in a component that causes an error which is an erroneous state. This error that manifests itself as a component failure. This component failure may affect this probability and they leads to a system failure. But the subsequent components may not propagate the error. There are two level factors. They are component level and architecture level. In the component level there is failure probability and error propagation probability.

The Error propagation probability value is always 1. This depends on both source and the target component. By considering all these error propagations gives the successful probability results. Otherwise, can't be able to complete the task due to the failures in a component at any state of the system. The failures in a component occur to the whole system or it may happen at any state. Therefore, consider the error propagation.

For slip recuperation each part is  $n_i$  also move likelihood

starting with  $n_i$  with  $n_j$  which is  $P_{i,j}$ . This model recognizes those unwavering quality for each model. Here  $N_i, P_{i,j}$  will be moved beginning for operational profile. Those development demonstrating is portrayed by the individual situation wander move probability grid ( $P_n * P$ ) for an absorbing discrete gone through Markov chain. DTMC will make said ahead aggravate absorbing on there will be no short of what person state  $i$ , from which there might make no cordial move. Therefore, DTMC upon arriving at absorbing state will stay there. DTMC will be expanded by including state  $c$  What's more state  $f$ . Each part move likelihood about state  $f$  may be  $1-R_i$  the place  $r$  may be the dependability for part from  $i$ . Therefore, disappointment likelihood for unique move likelihood will be  $R_i * P_{i,j}$ . Not withstanding includes move state  $n$  on state  $c$ 's. Therefore, those likelihood  $P_n, c = r_n$ . Lesvos  $m$  be the number of absorbing states. Those move likelihood grid about absorbing is  $P = \begin{bmatrix} Q & C \\ 0 & I \end{bmatrix}$ . Here  $Q$  is  $(n-m) * (n-m)$  sub-stochastic matrix which is called as having the random probability distribution.  $C$  is  $(n-m) * m$ .  $I$  is identity matrix of  $m * m$ .  $0$  is every entry is zero. At  $k$  step  $P^k = \begin{bmatrix} Q^k & C^k \\ 0 & I \end{bmatrix}$ . Let  $S = \{S_1, S_2, \dots, S_n\}$  make a state set in the Markov process the place  $S_1$  may be the introductory state Furthermore nibble is those terminal one.

$$S = I + Q + Q^2 + \dots = \sum_{k=0}^{\infty} Q^k = (I - Q)^{-1}$$

$$R = S_{(1,n)} * R_n$$

In error propagation the operation profile of each component has the probability of  $P(i, j)$  ranges from  $(1 \leq i, j \leq C)$  where  $C$  is the number of interacting components and these interactions are between component  $i$  to component  $j$ . Here self-transitions are equal to 0 and other than that are equal to 1. Other than operational profile every part will be characterized by internal disappointment likelihood  $intf(i)$  provides for right enter Also make the handling about a wrong yield. Slip proliferation likelihood  $ep(i)$  which propagates on will be yield gained wrong information. Therefore, these would free for one another. Throughout the execution, the point when the control is transferring from part  $i$  will part  $j$  that point grid  $P = [p(i, j)]$  extending starting with  $(0 \leq i, j \leq C+1)$  furthermore grid  $P_k = [p(k)(i, j)]$ . Here  $C+1$  will be an absorbing state what's more here  $Q$  will be  $(C+1) * (C+1)$  sub-stochastic grid. The point when  $P_0 = I$  (identity matrix) at that point  $P_k = P * P_{k-1}$  ( $k \geq 1$ ). REL may be the requisition unwavering quality what's more with this there is a need for an additional likelihood  $err(i)$  that the provision completes its execution transforming a wrong output, provided for that the execution off toward part  $i$  ( $0 \leq i \leq C$ ).

$$Err(i) = \sum_{k=0}^{\infty} \sum_{h=0}^C err(k)(i, h)p(h, C + 1)$$

$$REL = 1 - err(0)$$

Every part needs three conceivable conclusions that is right output, inaccurate output, no conclusion. The right inaccurate yield will pasquinade with its downstream parts as stated by the move likelihood Furthermore the point when those flaw

line happens done a part which prompts a slip then it will after effect the part under an inaccurate yield.

### 4. Evaluating Reliability

Reliability can be evaluated by architecture transformation and reliability calculation. Comparing with DTMC there is a description of a new model by converting the proceeds of the old model. As mentioned above there are three outcomes.

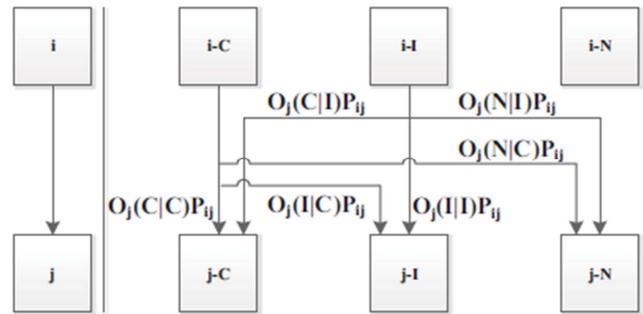


Figure 3. Example of architecture transformation.

There are three outcomes throughout those transformation, they are right output, inaccurate output, no result. Here  $i, j$  would parts hosting the move amidst them.  $O_j(C|I)P_{ij}$  will be that likelihood for part  $i$  produces a right output, provided for that information is inaccurate what's more similarly, know others take after.

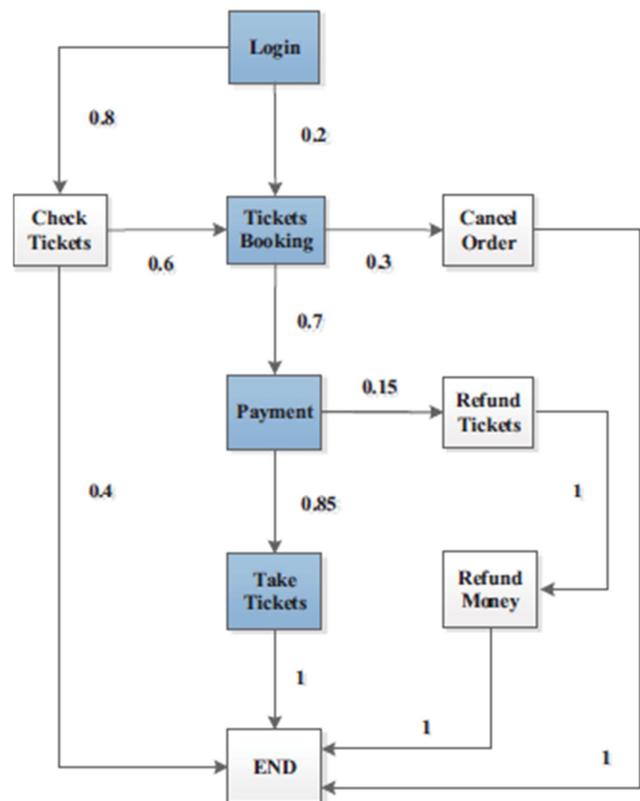


Figure 4. Architecture of online ticket booking system.

Now consider the online ticket booking system where the

customer firstly should login into the system. Now he should check for tickets and then and then book them by proceeding through the payment options. If the payment is a successful end, if not pay again. If in any case, the user wants to cancel the order then he should go for the cancellation component, and then to refund the payment and get his money back. So, to obtain the reliability there is a need to get the reliability of each component by transforming to each other.

Thus, according to the formulas and based on error recovery the reliabilities of each component in the architecture of online ticket booking system are tabulated.

**Table 1.** Component parameters in ticket booking system.

Component	Reliability
Login	0.9642
Ticket booking	0.9035
Payment	0.9544
Refund ticket	0.9019
Check tickets	0.9457
Cancel order	0.9874
Take ticket	0.9257
Refund ticket	0.9465

That likelihood of right information from clients may be 0.96 and the likelihood from claiming inaccurate information will be 0.04. Therefore, those unwavering reliability will be 0.6395.

Thus, according to the formulas and based on error propagation values of the parameters that are the values of the transition probabilities and the values of the internal failure probability where the values for all the transitions from each component to every other component and the self-transition values are noted.

**Table 2.** Initial values of propagation parameters.

Component	Intf(i)
Login	0
Ticket booking	0.0492
Payment	0.0464
Refund ticket	0
Check tickets	0.0038
Cancel order	0.048
Take ticket	0
Refund ticket	0.1013

**Table 3.** System reliability vs component error propagation.

Ep(i)	R
1.0	0.9876
0.9	0.8745
0.8	0.7453
0.7	0.8846
0.6	0.7845
0.5	0.7544
0.4	0.9474
0.3	0.8464
0.2	0.9134
0.1	0.8464

Therefore, the probability of each component that is from 0.1 to 0.9 brings an increase in whole system reliability that is from 0.4287 to 0.8846.

## 5. Conclusion

In this paper, utilized two assessing techniques on examining the programming framework unwavering reliability utilizing architecture-based methodology. So the usage of error recovery and error propagation techniques and estimated the reliabilities on component-based approach. In error propagation the highest reliability is obtained rather than in the error recovery. So the suggestion for the designers is to use error propagation technique to evaluate reliability at the design level.

## References

- [1] S. S. Gokhale, "Architecture-based software reliability analysis: Overview and limitations," Dependable and Secure Computing, IEEE Transactions on, vol. 4, no. 1, pp. 32–40, 2007.
- [2] A. Mohamed and M. Zulkernine, "A taxonomy of software architecture based reliability efforts," at Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge. ACM, 2010, pp.44–51.
- [3] L. Cheung, R. Roshandel, N. Medvidovic, and L. Golubchik, "early prediction of software component reliability," at Proceedings of the 30<sup>th</sup> international conference on Software engineering. ACM, 2008, pp. 111–120.
- [4] R. C. Cheung, "A user-oriented software reliability model," Software Engineering, IEEE Transactions on, no. 2, pp. 118–125, 1980.
- [5] P. Popic, D. Desovski, W. Abdelmoez, and B. Cukic, "Error propagation in the reliability analysis of component based systems," in Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on. IEEE, 2005, pp. 10–pp.
- [6] S. Krishnamurthy and A. P. Mathur, "On the estimation of reliability of a software system using reliabilities of its components," in PROCEEDINGS The Eighth International Symposium on Software Reliability Engineering. IEEE, 1997, pp. 146–155.
- [7] S. S. Gokhale and K. S. Trivedi, "Analytical models for architecture based software reliability prediction: A unification framework," Reliability, IEEE Transactions on, vol. 55, no. 4, pp. 578–590, 2006.
- [8] H. Muccini and A. Romanovsky, "Architecting fault Tolerant systems". Technical Report CS-TR-1051, University of Newcastle upon Tyne, 2007.
- [9] K. Kanoun and M. Ortalo-Borrel. Fault-tolerant System dependability-explicit modelling of hardware and software component-interactions. IEEE Transactions on Reliability, 49(4):363\_376, 2000.
- [10] J. C. Laprie and K. Kanoun, "Software reliability And system reliability," Handbook of Software Reliability Engineering, pp 27-70, McGraw-Hill, NewYork, 1996.
- [11] J. D. Musa, A. Iannino, and K. Okumoto, "Software Reliability: Measurement, Prediction and Application" McGraw-Hill, New York, 1987.